EXPERIENCES AND ENHANCEMENTS OF ReGTime:  
A SYSTEM FOR TRADING COMPUTING POWER

ANNJA ZAHN¹, THEO UNGERER²
¹University of Augsburg, Institute of Informatics, D–86135 Augsburg, Germany, Zahn@lnformatik.Uni-Augsburg.DE, phone +49–821–598–2176, fax +49–821–598–2200  

Abstract

ReGTime (Rent Gigaflops someTimes) is a software package for an envisioned “computing power market”. It helps customers to search for providers who lease workstation clusters, it manages the leasing of disposable machines on provider’s side and is responsible for granting access to rented systems and for compliance with established contracts. This paper concentrates on the experiences we have made while trying to establish the proposed computing power market with ReGTime in reality. We classify the problems as the lack of parallel applications outside of university research, the emotional component, the security problem, the problems of usability and insufficient information. The paper proposes solutions to all these problems.

Keywords: ReGTime, distributed computing, electronic commerce, computing power market

1. INTRODUCTION

Our vision is an electronic market for computing power, which allows trading and brokering of computing power in the Internet as market place. Such a new electronic market is able to adjust the state of imbalance between a surplus and a lack of computing capacity. Computing power becomes a commodity which is traded as any other commodity in electronic commerce.

Our software package ReGTime (Rent Gigaflops someTimes) [1, 2] is the first and to our knowledge the sole approach to realize a market for computing power. It is the idea of ReGTime that managers of networked computers offer their free capacity on the computing power market and customers are able to rent resources on a short-term basis. Therefore customers specify their requirements using World Wide Web. ReGTime creates an offer based on available capacities. If the offer is accepted, ReGTime helps to establish a contract, organizes the access, observes the compliance with the contract, and collects data for invoicing.

Four main problems have been crystallized from the experiences we have made by searching for users. A substantial point for a computing power market is that the customers
are not yet familiar with parallel and distributed computing. There are only few applicants on customers side for the new market. Second, potential providers are reluctant to let another person use their machine, because they are afraid to lose the availability of their machine for themselves. We characterize this problem as emotional acceptance. The third problem is the fear of misuse and intruders, and the last one is the insufficient information and usability of the ReGTime system as it is. We suggest solutions and show that our envisioned computing power market is substantiated and can become reality.

2. ELECTRONIC MARKET FOR COMPUTING POWER WITH ReGTime

The increasing demand for high performance computing is satisfied by parallel processing using multiprocessor systems. An alternative solution for coarse-grained parallel applications is distributed computing using workstation clusters instead of costly multiprocessors. Even the coupling of computers over large distances to a single virtual machine is possible and can be used to run resource-intensive applications in acceptable time. Recent research in this area aims at “metacomputing” or “virtual computer centers” [3], i.e. the cooperative use of several parallel computers by collaborating computer centers, and at “hypercomputing” [4], i.e. coupling of geographically distributed workstation clusters to a single computing resource.

An instrument to coordinate all actions and interactions for renting computing power is strongly demanded. Because of the nature of computing power the market place Internet is cut out for this commodity. In the concept of ReGTime, there are three classes of participants in the computing power market: providers, customers and brokers. To be part of the market, a provider offers his workstation cluster with ReGTime. The customer contacts a broker and asks for offers. The broker holds the information about providers willing to let their machines, thus, the broker acts as a mediator between customer and provider. The customer selects one of the offers, given as a reply from the broker. If the customer accepts an offer he has to conclude a contract with the provider. The broker himself is no partaker of that contract. Thereafter the customer is able to use the rented workstation cluster for distributed computing.

The customer is authorized to access the rented machines during the term of the contract. All customer activities are logged by the ReGTime software. These accounting informations are used to compute the invoice. ReGTime’s accounting mechanism is based on the standard Unix accounting system, which measures CPU-time (kernel and user mode), average memory consumption and I/O-activities of the customer’s processes. The prices for each category may be part of the contract. Alternatively, fixed costs for a contract and costs based solely upon the duration of a contract are also possible. After the period of lease, the account is closed and an invoice is sent to the customer. More detailed descriptions of the architecture of ReGTime and its realization can be found in [1, 2].

ReGTime fulfills the basic features of an electronic market, which is independent of space and time, the market itself is not stationary and therefore global. As point of departure to examine and judge ReGTime’s market mechanisms we have chosen the model of

24
market transactions which forms three phases of market transactions: information phase, agreement phase and processing phase.

To use ReGTime two basic preparations must be done by providers and customers. Providers have to register their clusters at a broker and implement the ReGTime software. An optional but desired requirement is passing the customers' public PGP-key to potential providers. After that, the information phase starts.

**Information phase**

In general this phase provides customers and providers with informations about e.g. the partners, products and services. Up to now ReGTime's information phase is scanty with information. It works as follows: The customer contacts a broker and asks for offers. Therefore the customer specifies his requirements which are at least the number of required computing nodes and a duration time. An offer includes a granted time period, the number of providers to satisfy the request and pricing informations if the offer can be fulfilled, otherwise the offer is rejected. In short the answer to the customer is not much more than "yes – available" or "no – not available". The customer has no possibility to find out what computing resources are disposable by the broker except by repeatedly asking for offers. The information phase ends with making a bid on providers side, which is processed from the broker to the described offer. Now the agreement phase begins.

**Agreement phase**

We implemented the agreement phase as a first-price sealed-bid auction. This means, every provider just has one chance for a bid which is offered through the broker to the customer. The customer has the choice to accept or reject the offer. No further negotiation occurs about the composition of machines, prices or terms of the contract. If customer and provider agree with each other, the contract has to be concluded. The optional but desired PGP-signature of the contract brings the agreement phase to an end. Now the customer gets all the information about the rented workstations and the transaction phase will start at contracting time.

**Transaction phase**

The job of the transaction phase is the fulfilment of the contract. At the moment ReGTime offers no comfortable tool for the usage of the remote resources. The customer itself must be familiar with the operating system and configuration of the rented cluster.

Figure 1 summarizes ReGTime's communication infrastructure during the proceeding and the results of the three phases of market transactions. In the next section we explain our experiences whilst putting ReGTime into practice and describe the problems which emerged.
3. EXPERIENCES AND PROBLEMS

3.1. Experiences

ReGTime is available on SUN workstations running Solaris 2.5.1 and on IBM RS/6000 workstations with operating system AIX 4.1. Portability turned out to be good, because the implementations of ReGTime's components use widespread programming mechanisms (e.g. PERL and C) only. Our experiences stem from the presentation of the ReGTime prototype at the CeBIT’96 in Hannover, from further activities to reach potential users within the following year, and from the participation in the project “hypercomputing with workstation clusters” at the University of Rostock.

The latter project [4] aimed at the organization of a hypercomputer consisting of workstation clusters of different institutions geographically distributed over Germany. Some applications were run on the hypercomputer which were of demonstrative nature, e.g. RSA-Challenge '97.

The demonstration system of ReGTime at the CeBIT'96 consisted of two independent workstation clusters at the universities of Augsburg (up to 20 IBM workstations) and Karlsruhe (four SparcStation5). The clusters were accessed from the exhibition, an additional IBM RS/6000 on the fair ground acted as a third cluster. ReGTime proved to be easy to use, access to computing power is provided fast and comfortably. We successfully demonstrated several existing distributed applications (based upon PVM) in conjunction with ReGTime.

Our next aim was to test ReGTime in industrial projects. However, we experienced a certain reluctance of potential providers.

3.2. Problems

We classify the lack of interest of industrial participants with ReGTime and the hypercomputing project due to the following four objections, which are illustrated in figure 2.
Figure 2: Problems in a computing power market

The lack of parallel applications

The availability of coarse-grained parallel programs that follow the message-passing paradigm is a prerequisite for the use of workstation clusters as parallel computers. Message-passing programming is wide-spread at universities as a programming method for parallel computers and distributed computing platforms. The parallel programming environments PVM (Parallel Virtual Machine) and MPI (Message Passing Interface) are based on the message-passing model, and allow portable programming of workstation clusters and parallel computers. Until now, using parallel PVM or MPI programs is mostly restricted to universities and large research institutions which already own costly parallel computers.

The emotional component

People are not willing to let strangers use their machine. They fear that their machine is not available when they need it. They do not like the idea of strangers to use their “personal” machine.

The security problem

Closely coupled with the emotional component are security objections despite the authentication mechanisms already provided by ReGTime. Our experiences showed that one of the most critical parts of acceptance of a computing power market are security aspects of prying and manipulating data. Through ReGTime, customers get access to conventional UNIX accounts. Using this account, a customer might be able to pry into the provider’s data. Considering the insufficient security properties of today’s computer systems granting access to an unknown customer is risky. This problem concerns both providers and customers.

Insufficient usability and information problems by ReGTime

So far ReGTime rents out raw UNIX logins. There is no help for the customer to get along with an unknown machine configuration and an unknown operating system. Moreover, if ReGTime generates an offer which is unacceptable for the potential customer,
the customer is not provided with further information that lets him adjust his demands to the available machines.

In the next section we propose solutions to all these problems and will focus on the respective enhancements of ReGTime.

4. SUGGESTED SOLUTIONS

Solutions to the lack of parallel applications

With ReGTime, companies are able to temporarily access high computing capacities without a high investment for a powerful computer system. We envision a stronger dissemination of parallel programs in the future fuelled by recent technological advances, by the evolution of new application domains where parallel applications will become more relevant to non-academic customers, and by the need for companies to apply qualitative better models. Such technological advances have produced network interfaces that provide users with low-latency communication hardware usable to build an efficient low-cost parallel computer from a workstation cluster (see e.g. Sun's S-Connect [5], DEC's Memory Channel and Dolphins SCI-Interconnect used in various research projects). Moreover, research on Software Distributed Shared Memory (DSM) systems develops efficient models to spread shared memory parallel programs over the machines of a workstation cluster (e.g. the Rthreads software [6] allows to spread POSIX threads over a workstation cluster). The application range is enhanced by Software DSM systems beyond the message-passing model reaching many new applications areas with already existing code written for multiprocessor workstations. Both together – low-cost low-latency network interfaces and Software DSM – will force more companies into the use of parallel code to hold a competitive market position.

Solutions to the emotional component of potential providers

The emotional component is totally irrational as far as it is not stimulated by security fears. The emotional component can be solved by better information and by appropriate software and organizational techniques. Fear of unavailability of the personal machine could be lessened if the renting out of machines is restricted to night times or weekends when the machine is never used by the provider.

Moreover in the near future we envision a less restrained emotional relationship between user and his machine. Load-sharing and load-balancing facilities like e.g. DQS [7], LSF, Condor, or LoadLeveler allow to use a workstation cluster as anonymous computing resource. A load-sharing facility provides users with machines that are appropriate for the users' needs observing load-balancing requirements. The owner of the machines are not interfered with their computing. Our main goal is therefore to couple ReGTime with load-balancing systems.

ReGTime originally aimed at a worldwide open computing power market. Fear of strangers can be faced by renting out only to people personally or institutionally known to the provider. That could be people within a large company or organized in an association. We therefore envision an application of ReGTime coming up first among partners.
which are not totally unknown to each other – like departments within a big company or business partners within a virtual corporation.

**Solutions to the security problem**

Because we cannot replace existing operating systems, we cannot significantly strengthen their protection and security mechanisms. One important precaution would be a machine administration that protects the computers with important data and programs through a firewall from the machines to let through ReGTime. The customer has the ability to protect his data actively, whilst the provider has to trust in the configuration of his general installation.

We stress the fact that authentication in ReGTime is already provided by PGP. So the user is known at least to the manager and security violations can be tracked back to the violating person. It is much more risky to lease computers to outsiders. A solid authentication mechanism increases the hindrance for prying and sabotaging and allows liability of harming. Upcoming certification centers allow to certify PGP key and further strengthen the confidence in the authentication process in near future. Security violations are therefore unlikely.

However, further provisions have to be made. One solution is the use of rented software only in restricted or confined shells, similar to the anonymous ftp shell. Thereby access is highly restricted. We are working towards a restricted shell script which is coupled with a better user guidance, see next section.

Another possibility would be the restriction of customer software to Java byte code. That would solve the security problem because Java byte code is verified before execution and violations are impossible as far as known. Moreover, Java byte code solves the problem of code reuse on a different machine and with different operating system settings as the one the code is developed and makes remote compilation unnecessary. However, up to now Java byte code is interpreted and therefore very slow and not appropriate for parallel high-speed applications. We do not support this approach in the near future. However, there exist a number of Java-restricted hypercomputing projects as e.g. Javeline, Jet, DAMP, Atlas, ParaWeb and Popcorn.

Another problem might be reluctance of providers to install the ReGTime software itself. Because ReGTime needs to establish and remove user accounts, it runs in the system space. It would be more acceptable if it could run in the user space. We cannot solve the problem totally but we are working towards a small kernel that needs to run in system space and the bulk of ReGTime to potentially run in user space. A more secure internet and more secure operating systems in the future might help to assert ReGTime-based concepts even in the envisioned worldwide open computing power market.

**Solutions to the usability problems of ReGTime**

The information phase of ReGTime will be improved to such a potential that users are able to extract prices and descriptions of available machines before asking for an offer.
Moreover, we are developing a WWW form for user guidance to allow easy access of the rented computing resources, both software and hardware. As a result customers do not need to know the operating system of the rented machine. Transfer of programs, compilation, production runs, control and transfer of results are dealt with by the WWW form.

5. CONCLUSIONS AND FUTURE WORK

We described the software package ReGTime that organizes an envisioned “computing power market”. ReGTime helps customers to search for providers who lease workstation clusters, it manages the leasing of disposable machines on provider’s side and establishes the contract with the customer. Finally, it is responsible for granting access to rented systems and for compliance with established contracts and supports secure authentication between business partners by PGP signatures.

The experiences with ReGTime in reality showed some problems described as the lack of parallel applications on customers’ side, the emotional receding on providers’ side, the security, usability and insufficient information problems. The paper proposed solutions to all these problems. We cannot pursue all solutions simultaneously, but we are working towards coupling of ReGTime to the load-sharing system DQS to avoid the emotional component, towards an improved information phase, and towards a WWW-based user guidance that eases both the usability and the security problem.

REFERENCES


